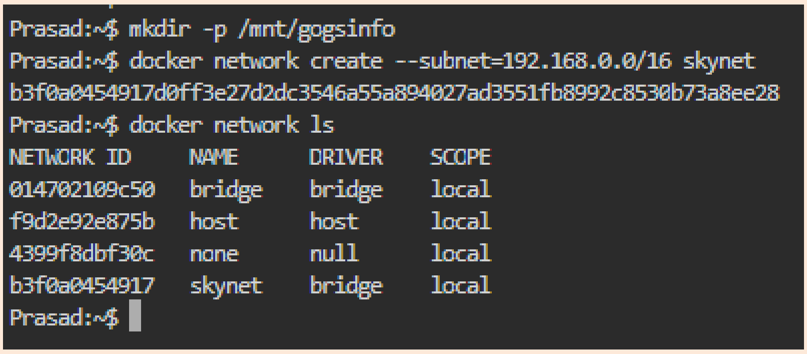
Tasks:

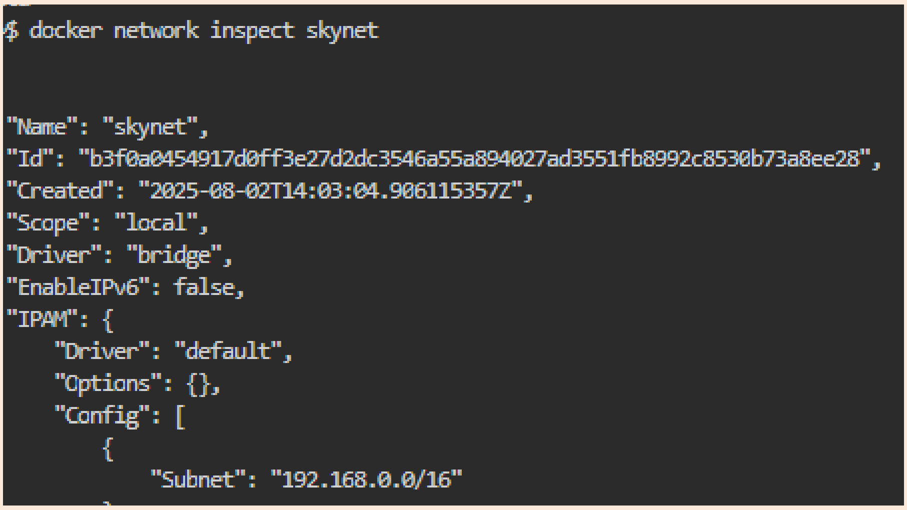
1. Create a multi-tier application of GOGS and POSTGRES.Create Gogs container by name 'mygitserver' and image 'gogs/gogs' and it should be accessible from port number 3000 of the host. Run a container with image postgres and name 'gogsdata', 'postgres' user's password should be 'redhat123' and, 'gogsdb' database should get created. Also create a directory gogsinfo in /mnt and attach it to /data/gogs/conf directory of the gogs container.
2. Create a new network with subnet '192.168.0.0/16' and name of this network should be 'skynet'.
3. Run a container using any image in the network 'skynet' with name 'customapp'.

Steps

1 Create the host directory for Gogs configuration and Create the skynet network

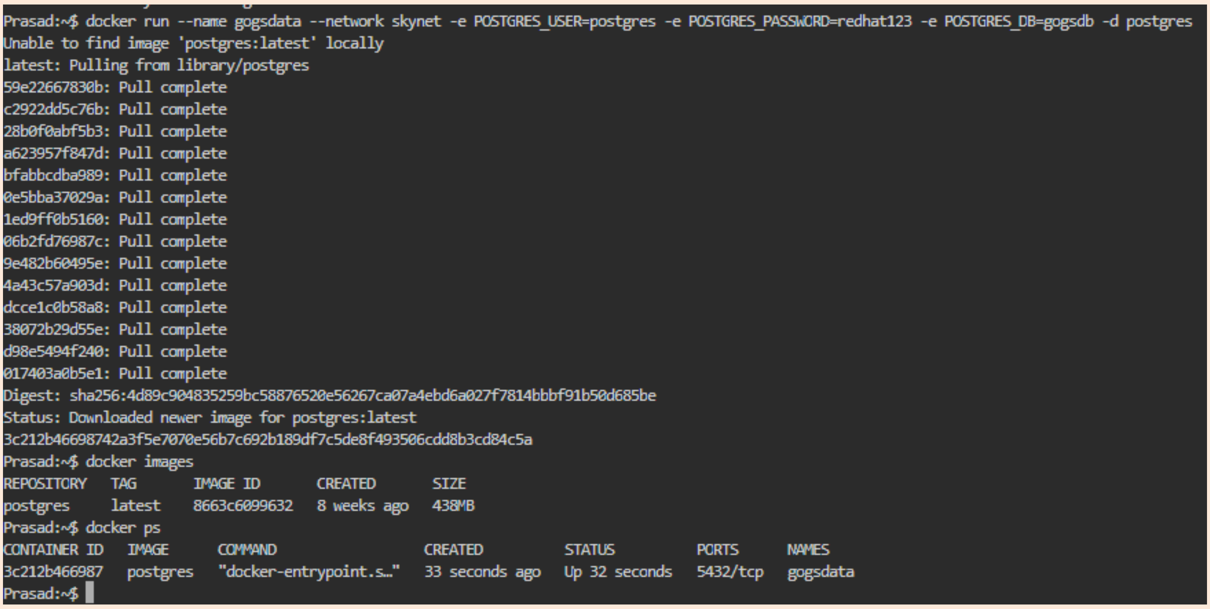


```
Prasad:~$ mkdir -p /mnt/gogsinfo
Prasad:~$ docker network create --subnet=192.168.0.0/16 skynet
b3f0a0454917d0ff3e27d2dc3546a55a894027ad3551fb8992c8530b73a8ee28
Prasad:~$ docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
014702109c50   bridge    bridge    local
f9d2e92e875b   host      host      local
4399f8dbf30c   none      null      local
b3f0a0454917   skynet    bridge    local
Prasad:~$
```



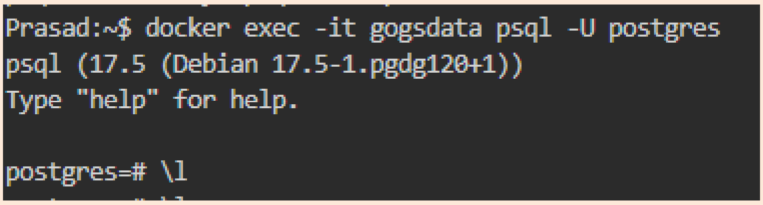
```
$ docker network inspect skynet


"Name": "skynet",
"Id": "b3f0a0454917d0ff3e27d2dc3546a55a894027ad3551fb8992c8530b73a8ee28",
"Created": "2025-08-02T14:03:04.906115357Z",
"Scope": "local",
"Driver": "bridge",
"EnableIPv6": false,
"IPAM": {
    "Driver": "default",
    "Options": {},
    "Config": [
        {
            "Subnet": "192.168.0.0/16"
```

: Create the PostgreSQL container on the skynet network



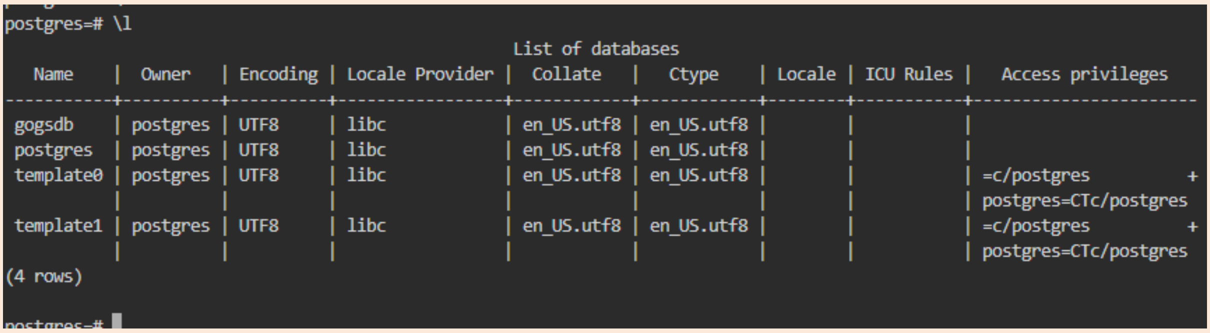
```
Prasad:~$ docker run --name gogsdata --network skynet -e POSTGRES_USER=postgres -e POSTGRES_PASSWORD=redhat123 -e POSTGRES_DB=gogsdb -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
59e22667830b: Pull complete
c2922dd5c76b: Pull complete
28b0f0abf5b3: Pull complete
a623957f847d: Pull complete
bfabbcdba989: Pull complete
0e5bba37029a: Pull complete
1ed9ff0b5160: Pull complete
06b2fd76987c: Pull complete
9e482b60495e: Pull complete
4a43c57a903d: Pull complete
dcce1c0b58a8: Pull complete
38072b29d55e: Pull complete
d98e5494f240: Pull complete
017403a0b5e1: Pull complete
Digest: sha256:4d89c904835259bc58876520e56267ca07a4ebd6a027f7814bbbf91b50d685be
Status: Downloaded newer image for postgres:latest
3c212b46698742a3f5e7070e56b7c692b189df7c5de8f493506cdd8b3cd84c5a
Prasad:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
postgres     latest    8663c6099632   8 weeks ago   438MB
Prasad:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS      NAMES
3c212b466987   postgres   "docker-entrypoint.s…"   33 seconds ago   Up 32 seconds   5432/tcp   gogsdata
Prasad:~$
```

Verification



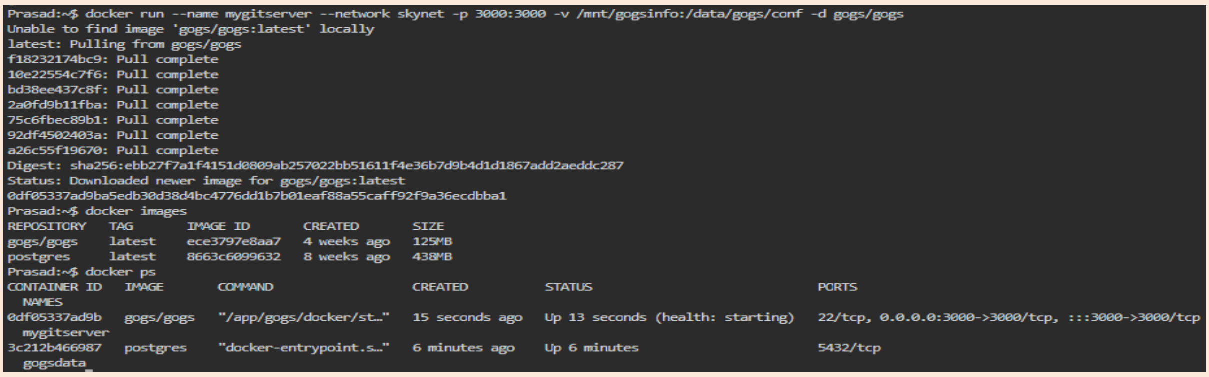
```
Prasad:~$ docker exec -it gogsdata psql -U postgres
psql (17.5 (Debian 17.5-1.pgdg120+1))
Type "help" for help.

postgres=# \l
```

List of databases inside the container



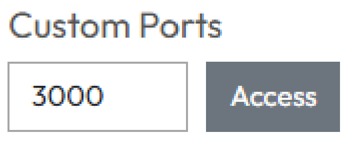
```
postgres=# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |  Collate   |   Ctype    | Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+------------+------------+--------+-----------+-----------------------
 gogsdb    | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |        |           |
 postgres  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |        |           |
 template0 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |        |           | =c/postgres          +
           |          |          |                 |            |            |        |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |        |           | =c/postgres          +
           |          |          |                 |            |            |        |           | postgres=CTc/postgres
(4 rows)

postgres=#
```

2. Create the Gogs container on the skynet network



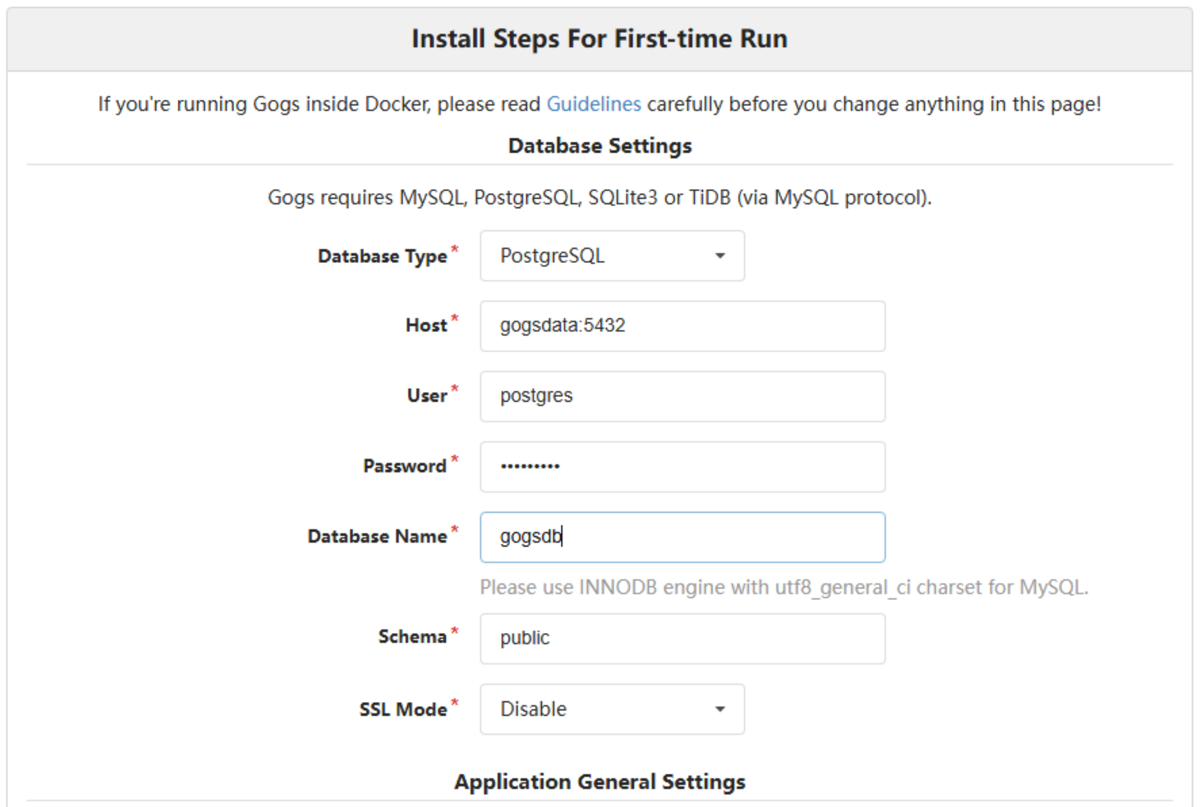
```
Prasad:~$ docker run --name mygitserver --network skynet -p 3000:3000 -v /mnt/gogsinfo:/data/gogs/conf -d gogs/gogs
Unable to find image 'gogs/gogs:latest' locally
latest: Pulling from gogs/gogs
f18232174bc9: Pull complete
10e22554c7f6: Pull complete
bd38ee437c8f: Pull complete
2a0fd9b11fba: Pull complete
75c6fbec89b1: Pull complete
92df4502403a: Pull complete
a26c55f19670: Pull complete
Digest: sha256:ebb27f7a1f4151d0809ab257022bb51611f4e36b7d9b4d1d1867add2aeddc287
Status: Downloaded newer image for gogs/gogs:latest
0df05337ad9ba5edb30d38d4bc4776dd1b7b01eaf88a55caff92f9a36ecdbba1
Prasad:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
gogs/gogs    latest    ece3797e8aa7   4 weeks ago   125MB
postgres     latest    8663c6099632   8 weeks ago   438MB
Prasad:~$ docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS                             PORTS
  NAMES
0df05337ad9b   gogs/gogs   "/app/gogs/docker/st…"   15 seconds ago   Up 13 seconds (health: starting)   22/tcp, 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
  mygitserver
3c212b466987   postgres    "docker-entrypoint.s…"   6 minutes ago    Up 6 minutes                       5432/tcp
  gogsdata
```

- Complete the Gogs web installation



Custom Ports

3000



Install Steps For First-time Run

If you're running Gogs inside Docker, please read Guidelines carefully before you change anything in this page!

Database Settings

Gogs requires MySQL, PostgreSQL, SQLite3 or TiDB (via MySQL protocol).

Database Type* PostgreSQL

Host* gogsdata:5432

User* postgres

Password* ••••••••••

Database Name* gogsdb

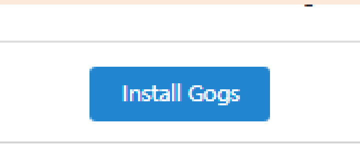
Please use INNODB engine with utf8_general_ci charset for MySQL.

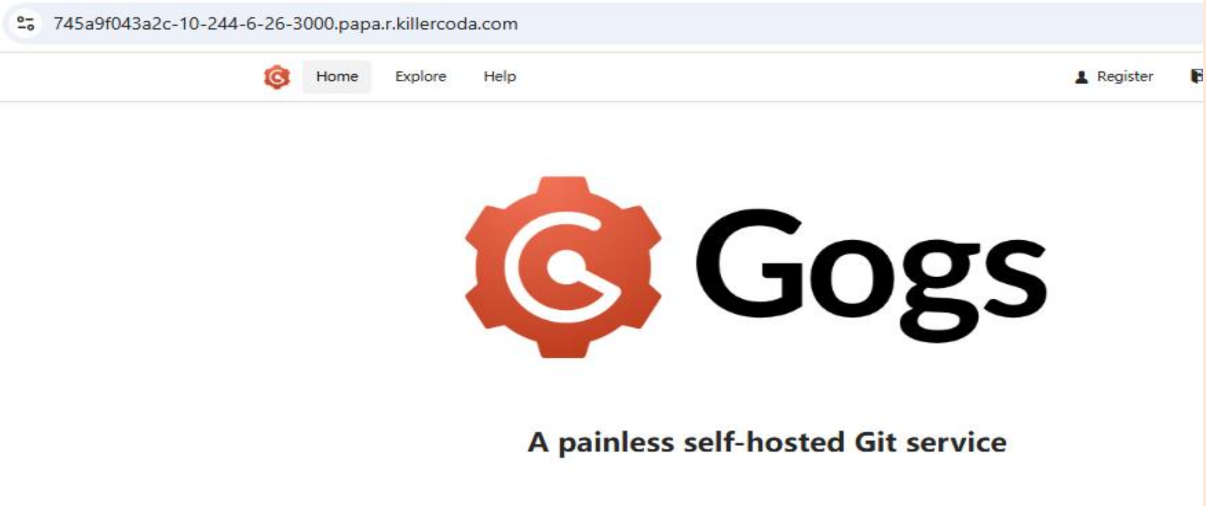
Schema* public

SSL Mode* Disable

Application General Settings

Click on install button

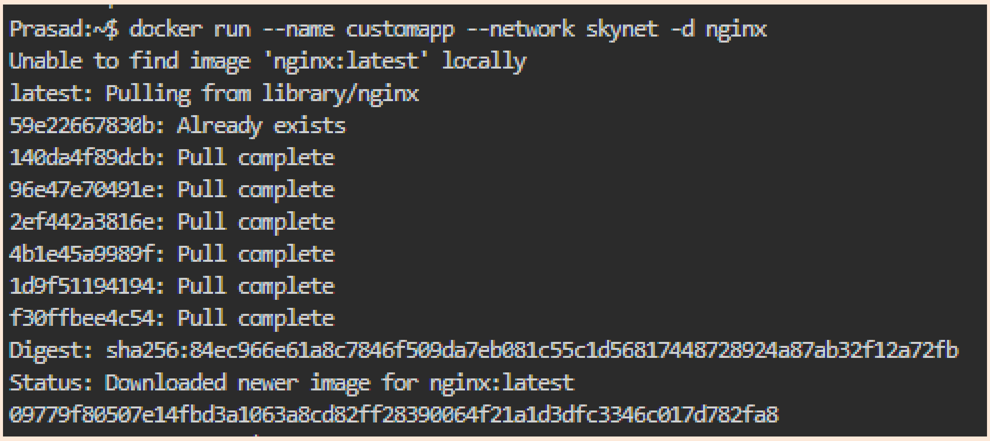




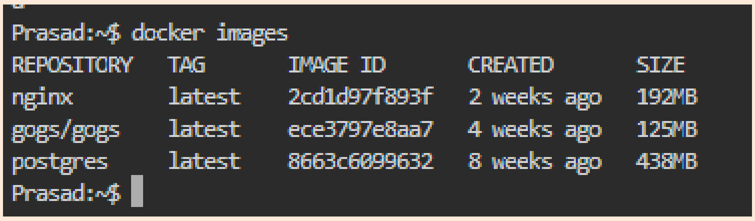
745a9f043a2c-10-244-6-26-3000.papa.r.killercoda.com

A painless self-hosted Git service

3. Run a Container on the Custom Network

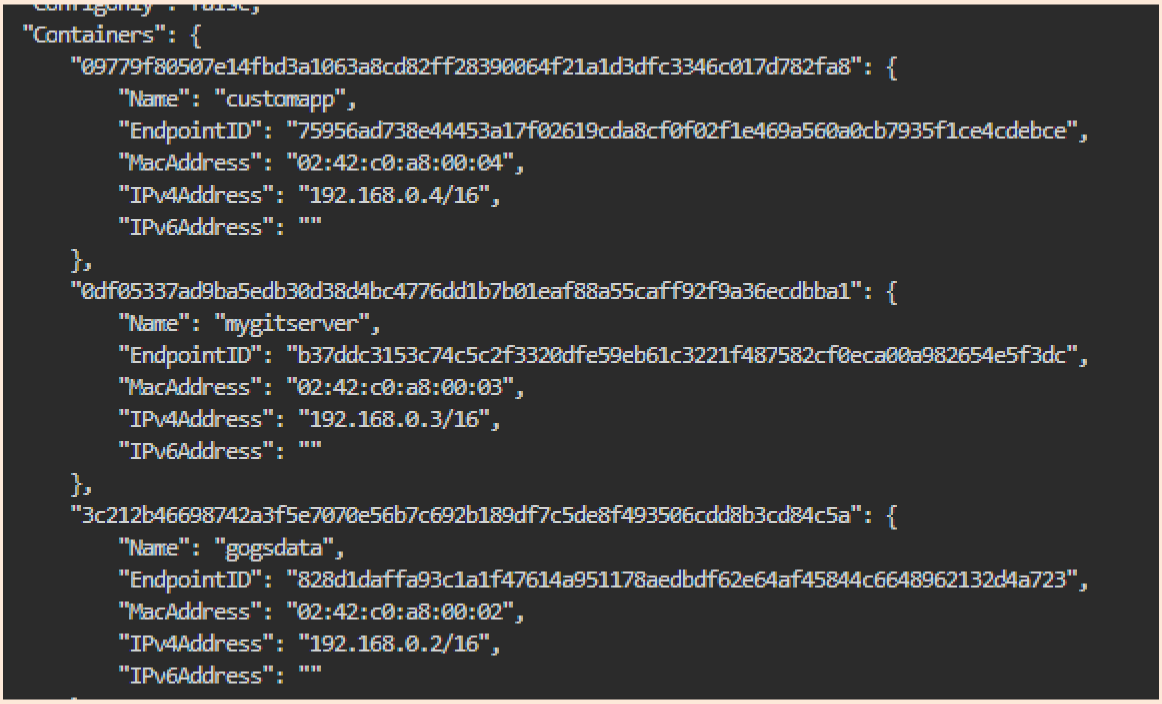


```
Prasad:~$ docker run --name customapp --network skynet -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
59e22667830b: Already exists
140da4f89dcb: Pull complete
96e47e70491e: Pull complete
2ef442a3816e: Pull complete
4b1e45a9989f: Pull complete
1d9f51194194: Pull complete
f30ffbee4c54: Pull complete
Digest: sha256:84ec966e61a8c7846f509da7eb081c55c1d56817448728924a87ab32f12a72fb
Status: Downloaded newer image for nginx:latest
09779f80507e14fbd3a1063a8cd82ff28390064f21a1d3dfc3346c017d782fa8
```



```
Prasad:~$ docker images
REPOSITORY   TAG      IMAGE ID       CREATED       SIZE
nginx        latest   2cd1d97f893f   2 weeks ago   192MB
gogs/gogs    latest   ece3797e8aa7   4 weeks ago   125MB
postgres     latest   8663c6099632   8 weeks ago   438MB
Prasad:~$ 
```

- Inspect network



```
"ConfigOnly": false,
"Containers": {
    "09779f80507e14fbd3a1063a8cd82ff28390064f21a1d3dfc3346c017d782fa8": {
        "Name": "customapp",
        "EndpointID": "75956ad738e44453a17f02619cda8cf0f02f1e469a560a0cb7935f1ce4cdebce",
        "MacAddress": "02:42:c0:a8:00:04",
        "IPv4Address": "192.168.0.4/16",
        "IPv6Address": ""
    },
    "0df05337ad9ba5edb30d38d4bc4776dd1b7b01eaf88a55caff92f9a36ecdbba1": {
        "Name": "mygitserver",
        "EndpointID": "b37ddc3153c74c5c2f3320dfe59eb61c3221f487582cf0eca00a982654e5f3dc",
        "MacAddress": "02:42:c0:a8:00:03",
        "IPv4Address": "192.168.0.3/16",
        "IPv6Address": ""
    },
    "3c212b46698742a3f5e7070e56b7c692b189df7c5de8f493506cdd8b3cd84c5a": {
        "Name": "gogsdata",
        "EndpointID": "828d1daffa93c1a1f47614a951178aedbdf62e64af45844c6648962132d4a723",
        "MacAddress": "02:42:c0:a8:00:02",
        "IPv4Address": "192.168.0.2/16",
        "IPv6Address": ""
```